Experiments in Newswire Summarisation

Stuart Mackie[1], Richard McCreadie[2], Craig Macdonald[2], and Iadh Ounis[2]

School of Computing Science, University of Glasgow, G12 8QQ, UK
[1]s.mackie.1@research.gla.ac.uk, [2]{firstname.lastname}@glasgow.ac.uk

Abstract. In this paper, we investigate extractive multi-document summarisation algorithms over newswire corpora. Examining recent findings, baseline algorithms, and state-of-the-art systems is pertinent given the current research interest in event tracking and summarisation. We first reproduce previous findings from the literature, validating that automatic summarisation evaluation is a useful proxy for manual evaluation, and validating that several state-of-the-art systems with similar automatic evaluation scores create different summaries from one another. Following this verification of previous findings, we then reimplement various baseline and state-of-the-art summarisation algorithms, and make several observations from our experiments. Our findings include: an optimised *Lead* baseline; indication that several standard baselines may be weak; evidence that the standard baselines can be improved; results showing that the most effective improved baselines are not statistically significantly less effective than the current state-of-the-art systems; and finally, observations that manually optimising the choice of anti-redundancy components, per topic, can lead to improvements in summarisation effectiveness.

1 Introduction

Text summarisation [15, 19] is an information reduction process, where the aim is to identify the important information within a large document, or set of documents, and infer an essential subset of the textual content for user consumption. Examples of text summarisation being applied to assist with user's information needs include search engine results pages, where snippets of relevant pages are shown, and online news portals, where extracts of newswire documents are shown. Indeed, much of the research conducted into text summarisation has focused on multi-document newswire summarisation. For instance, the input to a summarisation algorithm being evaluated at the Document Understanding Conference[1] or Text Analysis Conference[2] summarisation evaluation campaigns is often a collection of newswire documents about a news-worthy event. Further, research activity related to the summarisation of news-worthy events has recently been conducted under the TREC Temporal Summarisation Track[3]. Given the current research interest in event summarisation [5, 8, 13], the reproduction, validation, and generalisation of findings from the newswire summarisation literature is important to the advancement of the field, and additionally, constitutes good scientific practice.

Hence, in this contribution, we begin by reproducing and validating two previous findings, over DUC 2004 Task 2. First, that the ROUGE-2 [9] metric is aligned with user judgements for summary quality, but generalising this finding in the context of crowd-sourcing. Second, that there exists measurable variability in the sentence selection behaviour of state-of-the-art summarisation algorithms exhibiting similar ROUGE-2 scores, but confirming such variability via a complementary form of analysis, adding

[1] duc.nist.gov [2] nist.gov/tac [3] trec-ts.org

to the weight of evidence of the original finding. Further, in this paper, we reproduce the *Random* and *Lead* baselines, over the DUC 2004 and TAC 2008 newswire summarisation datasets. Observations from such experiments include: a validation of the lower-bound on acceptable summarisation effectiveness; findings that the effectiveness of the simple *Lead* baseline used at DUC and TAC can be improved; and that the *Lead* baseline augmented with anti-redundancy components is competitive with several standard baselines, over DUC 2004. Finally, we reproduce a series of standard and state-of-the-art summarisation algorithms. Observations from these experiments include: optimisations to several standard baselines that improve effectiveness; results indicating that state-of-the-art techniques, using integer linear programming and machine learning, are not always more effective than simple unsupervised techniques; and additionally, that an oracle system optimising the selection of different anti-redundancy components, on a per-topic basis, can potentially lead to improvements in summarisation effectiveness.

The remainder of this paper is organised as follows: We report our experimental setup in Section 2, describing summarisation algorithms, datasets, and the evaluation process. In Section 3, we present the results from a user study reproducing and validating previous findings that the ROUGE-2 metric aligns with user judgements for summary quality. In Section 4, we reproduce and validate previous findings that, despite exhibiting similar ROUGE-2 scores, state-of-the-art summarisation algorithms vary in their sentence selection behaviour. In Section 5, we reproduce the *Random* and *Lead* baselines, making several observations over the DUC 2004 and TAC 2008 datasets. In Section 6, we reproduce standard baselines and state-of-the-art systems, making further observations over the DUC 2004 and TAC 2008 datasets. Finally, Section 7 summarises our conclusions.

2 Reproducible Experimental Setup

In this section, we briefly describe the summarisation algorithms that we investigate, with full details available in the relevant literature [7]. Then, we also describe the anti-redundancy components that aim to minimise repetition in the summary text. Following this, we provide details of the evaluation datasets and metrics used in our experiments.

Summarisation Algorithms – In general, each summarisation algorithm assigns scores to sentences, computing a ranked list of sentences where the highest-scoring sentences are most suitable for inclusion into the summary text. Some algorithms then pass the ranked list of scored sentences to an anti-redundancy component, described below, while other algorithms do not (i.e. handling redundancy internally). **FreqSum** [16] computes the probability of each word, over all the input sentences. Sentences are scored by summing the probabilities of each of its individual words, normalising by sentence length (i.e. average probability). The scored sentences (a ranked list) are passed to an anti-redundancy component for summary sentence selection. **TsSum** [2] relies on the computation of topic words [10], which are words that occur more often in the input text than in a large background corpus. The log-likelihood ratio test is applied, with a threshold parameter used to determine topic words from non-topic words. A further parameter of this algorithm is the background corpus to use; in our experiments we use the term frequencies of the 1,000,000 most common words in Wikipedia. Sentences are scored by taking the ratio of unique topic words to unique non-topic words. An anti-redundancy component is then applied to select novel sentences. **Centroid** [18] computes a centroid pseudo-document of all terms, and scores sentences by their cosine similarity to

this centroid vector. This algorithm has a parameter, in that a vector weighting scheme must be chosen, e.g. *tf***idf*. Sentence selection is via an anti-redundancy component. **LexRank** [3] computes a highly-connected graph, where the vertices are sentences, and weighted edges represent the cosine similarity between vertices. Again, a vector weighting scheme, to represent sentences as vectors, must be chosen. Sentences are scored by using a graph algorithm (e.g. in-degree or PageRank) to compute a centrality score for each vertex. A threshold parameter is applied over the graph, disconnecting vertices that fall below a given cosine similarity, or, the edge weights may be used as transition probabilities in PageRank (i.e. *Cont. LexRank*). Further, an anti-redundancy component is then used to select novel sentences. **Greedy–KL** [6] computes the Kullback–Leibler divergence between each individual sentence and all other sentences. Then, summary sentences are chosen by greedily selecting the sentence that minimises the divergence between the summary text and all the original input sentences. This algorithm has a parameter, in the range $[0,1]$, the Jelinek–Mercer smoothing λ value, used when computing the language models for the Kullback–Leibler divergence computation. **ICSISumm** [4] views the summarisation task as an optimisation problem, with a solution found via integer linear programming. An objective function is defined that maximises the presence of weighted concepts in the final summary text, where such concepts are computed over the set of input sentences (specifically, bi-grams valued by document frequency). In our experiments, we use an open source solver[1] to express and execute integer linear programs. Further, we also experiment with a **machine learned model**. The features used are the *FreqSum*, *TsSum*, *Centroid*, *LexRank*, and *Greedy–KL* baselines. The learned model is trained on the gold-standard of DUC 2002 (manual sentence extracts), and tested on DUC 2004 and TAC 2008. For our experiments, we train a maximum entropy binary classifier[2], with feature values scaled in the range $[-1,1]$. The probability estimates output from the classifier are used to score the sentences, producing a ranking of sentences that is passed through an anti-redundancy component for summary sentence selection.

Anti-redundancy Components – Each anti-redundancy component takes as input a list of sentences, previously ranked by a summarisation scoring function. The first, highest-scoring, sentence is always selected. Then, iterating down the list, the next highest-scoring sentence is selected on the condition that it satisfies a threshold. We experiment with the following anti-redundancy thresholding components, namely *NewWordCount*, *NewBigrams*, and *CosineSimilarity*. **NewWordCount** [1] only selects the next sentence in the list, for inclusion into the summary text, if that sentence contributes n new words to the summary text vocabulary. In our experiments, the value of n, the new word count parameter, ranges from $[1,20]$, in steps of 1. **NewBigrams** only selects a sentence if that sentence contributes n new bi-grams to the summary text vocabulary. In our experiments, the value of n, the new bi-grams parameter, ranges from $[1,20]$, in steps of 1. The **CosineSimilarity** thresholding component only selects the next sentence if that sentence is sufficiently dis-similar to all previously selected sentences. In our experiments, the value of the cosine similarity threshold ranges from $[0,1]$ in steps of 0.05. As cosine similarity computations require a vector representation of the sentences, we experiment with different weighting schemes, denoted *Tf*, *Hy*, *Rt*, and *HyRt*. *Tf* is textbook *tf***idf*, specifically $\log(\text{tf}) * \log(\text{idf})$, where *tf* is the frequency of a term in a sentence, and *idf* is

[1] gnu.org/software/glpk/ [2] mallet.cs.umass.edu/api/cc/mallet/classify/MaxEnt.html

N/Nt, the number of sentences divided by the number of sentences containing the term *t*. *Hy* is a *tf***idf* variant, where the *tf* component is computed over all sentences combined into a pseudo-document, with *idf* computed as N/Nt. *Rt* and *HyRt* are *tf***idf* variants where we do not use log smoothing, i.e. raw *tf*. The 4 variants of weighting schemes are also used by *Centroid* and *LexRank*, to represent sentences as weighted vectors.

Summarisation Datasets – In our summarisation experiments, we use newswire documents from the Document Understanding Conference (DUC) and the Text Analysis Conference (TAC). Each dataset consists of a number of topics, where a topic is a cluster of related newswire documents. Further, each topic has a set of gold-standard reference summaries, authored by human assessors, to which system-produced summaries are compared in order to evaluate the effectiveness of various summarisation algorithms. The DUC 2004 Task 2 dataset has 50 topics of 10 documents per topic, and 4 reference summaries per topic. The TAC 2008 Update Summarization Task dataset has 48 topics, and also 4 reference summaries per topic. For each topic within the TAC dataset, we use the 10 newswire articles from document set A, and the 4 reference summaries for document set A, ignoring the update summarisation part of the task (set B). Further, we use the TAC 2008 dataset for generic summarisation (ignoring the topic statements).

The Stanford CoreNLP toolkit is used to chunk the newswire text into sentences, and tokenise words. Individual tokens are then subjected to the following text processing steps: Unicode normalisation (NFD[1]), case folding, splitting of compound words, removal of punctuation, Porter stemming, and stopword removal (removing the 50 most common English words[2]). When summarising multiple documents for a topic, we combine all sentences from the input documents for a given topic into a single virtual document. The sentences from each document are interleaved one-by-one in docid order, and this virtual document is given as input to the summarisation algorithms.

Summarisation Evaluation – To evaluate summary texts, we use the ROUGE [9] evaluation toolkit[3], measuring n-gram overlap between a system-produced summary and a set of gold-standard reference summaries. Following best practice [7], the summaries under evaluation are subject to stemming, stopwords are retained, and we report ROUGE-1, ROUGE-2 and ROUGE-4 recall – measuring uni-gram, bi-gram, and 4-gram overlap respectively – with results ordered by ROUGE-2 (in bold), the preferred metric due to its reported agreement with manual evaluation [17]. Further, for all experiments, summary lengths are truncated to 100 words. The ROUGE parameter settings used are: "`ROUGE-1.5.5.pl -n 4 -x -m -l 100 -p 0.5 -c 95 -r 1000 -f A -t 0`". For summarisation algorithms with parameters, we learn the parameter settings via a five-fold cross validation procedure, optimising for the ROUGE-2 metric. Statistical significance in ROUGE results is reported using the paired Student's t-test, 95% confidence level, as implemented in MATLAB. ROUGE results for various summarisation systems are obtained using SumRepo [7][4], which provides the plain-text produced by 5 standard baselines, and 7 state-of-the-art systems, over DUC 2004. Using this resource, we compute ROUGE results, over DUC 2004 only, for the algorithms available within SumRepo, obtaining reference results for use in our later experiments.

---

[1] docs.oracle.com/javase/8/docs/api/java/text/Normalizer.html

[2] en.wikipedia.org/wiki/Most_common_words_in_English

[3] www.berouge.com

[4] www.seas.upenn.edu/~nlp/corpora/sumrepo.html

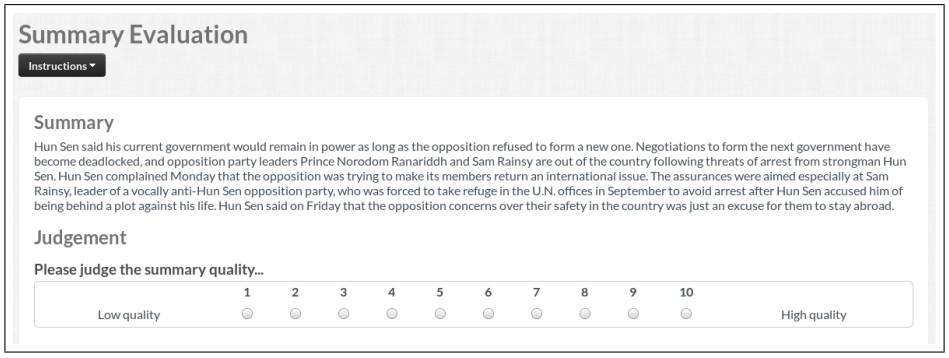

Fig. 1: The interface for our user study, soliciting summary judgements via CrowdFlower.

3 Crowd-sourced User Study to Validate that the ROUGE-2 Metric Aligns with User Judgements of Summary Quality

Current best practice in summarisation evaluation [7] is to report ROUGE results using ROUGE-2 as the preferred metric, due to the reported agreement of ROUGE-2 with manual evaluation [17]. In this section, we now examine if the ROUGE-2 metric aligns with user judgements, reproducing and validating previous findings – but generalising to the context of crowd-sourcing. This provides a measure of confidence in using crowd-sourced evaluations of newswire summarisation, as has previously been demonstrated for microblog summarisation [11, 12]. Our user study is conducted via CrowdFlower[1], evaluating 5 baseline systems and 7 state-of-the-art systems, over the DUC 2004 dataset using summary texts from SumRepo. A system ranking based on ROUGE-2 effectiveness is compared with a system ranking based on the crowd-sourced user judgements, in order to determine if the ROUGE-2 metric is aligned with user judgements.

Users are shown a summary text, and asked to provide a judgement on the quality of the summary, using a 10-point scale. The interface for soliciting summary quality assessments is shown in Figure 1. Users are provided with minimal instructions, which they may opt to read, and although we provide criteria by which users could make judgements of summary quality[2], we make no attempt to simulate a complex work task. The total cost of the user study is $109.74, for 3,000 judgements (50 topics, 12 systems, each summary judged 5 times, approx. $0.036 per judgement). The per-system judgements provided by the users are aggregated first at the topic level (over 5 assessors) and then at the system level (over 50 topics). Table 1 provides results from the user study, where we compare a ranking of systems based on their ROUGE-2 effectiveness (denoted *Reference Results*) with a ranking of systems obtained from the mean of the 10-point scale user judgements (denoted *User Judgements*). Table 1 also includes the ROUGE-1 and ROUGE-4 scores of each system for the reference results, and the minimum, maximum, and median scores for the user judgements. The 12 systems under evaluation are separated into two broad categories [7], namely *Baselines* and *State-of-the-art*.

From Table 1, we observe that, generally, the crowd-sourced user judgements mirror the ROUGE-2 system ordering of baselines and state-of-the-art systems, i.e. that it is therefore possible for the crowd to distinguish between baseline algorithms and state-of-the-art systems. The two exceptions are *CLASSY 04*, which the crowd-sourced user

[1] crowdflower.com [2] www-nlpir.nist.gov/projects/duc/duc2007/quality-questions.txt

Table 1: Reference ROUGE results, over DUC 2004, and results from our crowd-sourced user study validating ROUGE-2 is aligned with user judgements for summary quality.

Reference Results (*SumRepo*)				
Baselines	**Rank**	R-1	**R-2**	R-4
Cont. LexRank	1	36.00	7.51	0.83
Centroid	2	36.42	7.98	1.20
FreqSum	3	35.31	8.12	1.00
TsSum	4	35.93	8.16	1.03
Greedy–KL	5	38.03	8.56	1.27
State-of-the-art	**Rank**	R-1	**R-2**	R-4
CLASSY 04	6	37.71	9.02	1.53
CLASSY 11	7	37.21	9.21	1.48
Submodular	8	39.23	9.37	1.39
DPP	9	39.84	9.62	1.57
OCCAMS_V	10	38.50	9.75	1.33
RegSum	11	38.60	9.78	1.62
ICSISumm	12	38.44	9.81	1.74

User Judgements (*CrowdFlower*)					
Baselines	**Rank/**	**mean**	min	max	median
FreqSum	3	7.16	3.40	9.00	7.40
CLASSY 04	6	7.36	5.00	9.40	7.40
TsSum	4	7.60	5.00	9.20	7.60
Centroid	2	7.64	5.00	9.40	7.60
LexRank	1	7.66	2.60	9.60	7.80
State-of-the-art	**Rank/**	**mean**	min	max	median
OCCAMS_V	10	7.70	3.80	9.60	7.90
CLASSY 11	7	7.71	5.20	9.20	7.80
Submodular	8	7.75	5.60	9.40	7.80
DPP	9	7.80	5.20	9.80	8.00
RegSum	11	7.85	6.00	9.60	8.00
Greedy–KL	5	8.05	3.80	9.60	8.20
ICSISumm	12	8.10	5.60	9.80	8.20

judgements have rated less effective than the ROUGE-2 result, and *Greedy–KL*, which the crowd-sourced user judgements have rated more effective than the ROUGE-2 result. However, from Table 1, we can conclude that the ROUGE-2 metric is generally aligned with crowd-sourced user judgements, reproducing and validating previous findings [17], and generalising to the context of crowd-sourced summarisation evaluations.

4 Confirming Variability in Sentence Selection Behaviour of Summarisation Algorithms with Similar ROUGE-2 Scores

It has been previously reported [7], over DUC 2004 Task 2, that the top 6 state-of-the-art summarisation algorithms exhibit low overlap in the content selected for inclusion into the summary text, despite having no statistically significant difference in ROUGE-2 effectiveness (two-sided Wilcoxon signed-rank, 95% confidence level). Content overlap was measured at the level of sentences, words, and summary content units, demonstrating that the state-of-the-art algorithms exhibit variability in summary sentence selection. In this section, we seek to reproduce and validate this finding, by investigating the variation in ROUGE-2 effectiveness of the state-of-the-art systems across topics. This analysis seeks to determine if, despite having very similar ROUGE-2 effectiveness, the sentence selection behaviour of the state-of-the-art systems varies over topics. This would confirm that the state-of-the-art systems are selecting different content for inclusion into the summary, reproducing and validating the previously published [7] results.

For our analysis, we examine the ROUGE-2 effectiveness of the state-of-the-art systems over the 50 topics of DUC 2004 Task 2, using the summary text from SumRepo. In Figure 2, we visualise the distribution of ROUGE-2 scores over topics, for the top 6 state-of-the-art systems, with the topics on the x-axis ordered by the ROUGE-2 effectiveness of *ICSISumm*. In Table 2, we then quantify the ROUGE-2 effectiveness between the top 6 state-of-the-art systems, showing the Pearson's linear correlation coefficient of ROUGE-2 scores across the topics.

From Figure 2, we observe that, for each of the top 6 state-of-the-art systems, there is variability in ROUGE-2 scores over different topics. Clearly, for some topics, certain systems are more effective, while for other topics, other systems are more effective. This

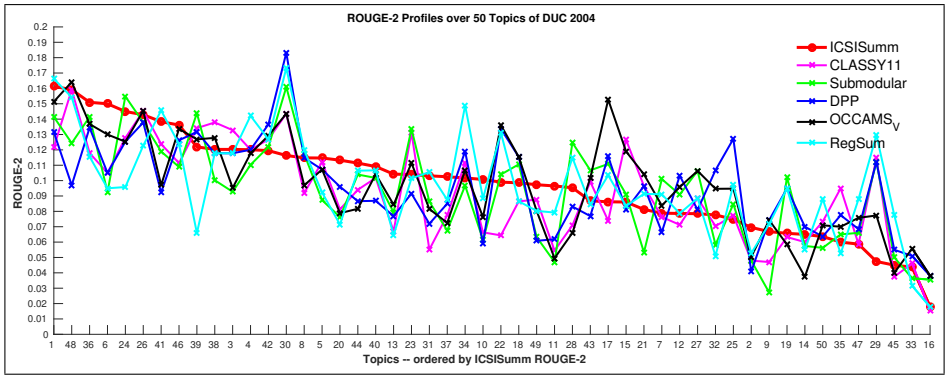

Fig. 2: ROUGE-2 effectiveness profiles, across the 50 topics of DUC 2004, for the top 6 state-of-the-art systems, with the x-axis ordered by the ROUGE-2 effectiveness of *ICSISumm*.

Table 2: Pearson's linear correlation coefficient of ROUGE-2 scores between the top six state-of-the-art systems, across the 50 topics of DUC 2004.

	CLASSY11	*Submodular*	*DPP*	*OCCAMS_V*	*RegSum*	*ICSISumm*
CLASSY11	1.0000	–	–	–	–	–
Submodular	0.7607	1.0000	–	–	–	–
DPP	0.6950	0.7605	1.0000	–	–	–
OCCAMS_V	0.7701	0.7456	0.7824	1.0000	–	–
RegSum	0.6721	0.7089	0.6849	0.6599	1.0000	–
ICSISumm	0.7385	0.6875	0.6089	0.7463	0.6516	1.0000

variability is usually masked behind the ROUGE-2 score, which provides an aggregated view over all topics. Further, from Table 2 we observe that the per-topic ROUGE-2 scores of the top 6 state-of-the-art systems are not as highly correlated as indicated by these system's aggregated ROUGE-2 scores, which have no statistically significant difference. Indeed, we observe from Table 2 that the highest level of correlation is 0.7824, between *OCCAMS_V* and *DPP*, but falls to 0.6089, between *ICSISumm* and *DPP*. Given the visualisation of variability in Figure 2, and the quantification of variability in Table 2, we conclude that, although these systems have very similar ROUGE-2 scores, they exhibit variability in sentence selection behaviour, validating the previous findings [7].

5 Reproducing the Random and Lead Baselines

In this section, we reproduce the *Random* and *Lead* baselines, making observations over DUC 2004 and TAC 2008. The *Random* baseline provides a lower-bound on acceptable effectiveness, i.e. an effective summarisation algorithm should out-perform a randomly generated summary. In our experiments, we generate 100 random summaries, per topic, and average the ROUGE-1, ROUGE-2 and ROUGE-4 scores to provide a final *Random* baseline result. The *Lead* baseline is reported to be very effective for newswire summarisation [14], due to journalistic convention of a news article's first sentence being very informative. We investigate the method used to derive the *Lead* baseline, and further, the results of augmenting the *Lead* baseline with different anti-redundancy components.

Table 3 gives the ROUGE results for the *Random* baseline, 2 variants of the *Lead* baseline (*recent doc* and *interleaved*), the *Lead (interleaved)* baseline passed through 6 anti-redundancy components, and also the results for the 5 standard baseline algorithms. In particular, Table 3 presents the ROUGE results for the *Lead (recent doc)* baseline used

Table 3: ROUGE scores, over DUC 2004 and TAC 2008, for Random and Lead, the Lead baseline augmented with different anti-redundancy components, and 5 standard baselines.

DUC 2004			
Lower-bound	R-1	**R-2**	R-4
Random	30.27	4.33	0.35
Lead (baselines)	R-1	**R-2**	R-4
Lead (recent doc)	31.46	6.13	0.62
Lead (interleaved)	34.23†	**7.66**†	1.18†
Lead (anti-redundancy)	R-1	**R-2**	R-4
CosineSimilarityRt	35.67‡	7.91	1.20
CosineSimilarityTf	36.02‡	7.97	1.20
NewWordCount	35.54‡	8.02	1.22
CosineSimilarityHyRt	35.91‡	8.08‡	1.24
NewBigrams	36.05‡	8.11	1.18
CosineSimilarityHy	36.38‡	**8.29**‡	1.29
Baselines (*SumRepo*)	R-1	**R-2**	R-4
LexRank	36.00	7.51	0.83✗
Centroid	36.42	7.98	1.20
FreqSum	35.31	8.12	1.00
TsSum	35.93	8.16	1.03
Greedy–KL	38.03✔	**8.56**	1.27

TAC 2008			
Lower-bound	R-1	**R-2**	R-4
Random	29.75	4.60	0.57
Lead (baselines)	R-1	**R-2**	R-4
Lead (recent doc)	29.73	5.83	0.79
Lead (interleaved)	33.18†	**7.69**†	1.44†
Lead (anti-redundancy)	R-1	**R-2**	R-4
CosineSimilarityHy	33.71	7.53	1.33
CosineSimilarityRt	33.44	7.70	1.41
CosineSimilarityTf	33.76	7.78	1.44
NewBigrams	33.92	7.87	1.43
NewWordCount	33.73	7.92	1.66
CosineSimilarityHyRt	34.08‡	**8.10**‡	1.55

**SumRepo* baselines are not available over TAC 2008

for the DUC and TAC evaluations, which consists of the lead sentences extracted from the most recent document in the collection of documents for a topic. We also show, in Table 3, the *Lead (interleaved)* baseline that results from the sentence interleaving of a virtual document, where the input sentences are arranged one-by-one from each document in turn. Further, Table 3 provides reference ROUGE results, over DUC 2004, for the 5 standard baselines computed using SumRepo (not available for TAC 2008).

From Table 3, we first observe the ROUGE effectiveness of the *Random* baseline, establishing a lower-bound on the acceptable performance over the two datasets. All of the standard baselines exceed the *Random* performance, however, *Lead (recent doc)* over TAC 2008 exhibits a ROUGE-1 score that is not significantly different from *Random*. This indicates that *Lead (recent doc)* may not be a strong baseline, over TAC 2008. Indeed, we observe a significant improvement in ROUGE results, shown in Table 3 using the "†" symbol, for the *Lead (interleaved)* baseline over the official *Lead (recent doc)* baselines used at DUC and TAC. From this, we conclude that using multiple lead sentences, from multiple documents, to construct a *Lead* baseline is more effective than simply using the first *n* sentences from the most recent document.

Further, from Table 3, we observe cases where the *Lead (interleaved)* baseline, when passed through an anti-redundancy component, achieves ROUGE effectiveness scores that exhibit a significant improvement over the *Lead (interleaved)* baseline, as indicated by the "‡" symbol. In particular, over DUC 2004, *Lead (interleaved)* augmented with anti-redundancy filtering results in significant improvements in ROUGE-1 scores for all anti-redundancy components investigated, and significant improvements in ROUGE-2 scores using *CosineSimilarityHyRt* and *CosineSimilarityHy*. However, from Table 3, we observe that anti-redundancy filtering of *Lead (interleaved)* is not as effective over TAC 2008, where only *CosineSimilarityHyRt* exhibits significantly improved ROUGE-1 and ROUGE-2 scores. From these observations, we conclude that the optimal *Lead* baseline,

for multi-document extractive newswire summarisation, can be derived by augmenting an interleaved *Lead* baseline with anti-redundancy filtering (such as cosine similarity).

Finally, from Table 3, we observe the 5 standard baselines, *LexRank*, *Centroid*, *FreqSum*, *TsSum*, and *Greedy–KL*, do not exhibit significant differences in ROUGE-2 scores, over DUC 2004, from *CosineSimilarityHy*, the most effective anti-redundancy processed interleaved *Lead* baseline. Indeed, only *Greedy–KL* exhibits a ROUGE-1 score ("✔") that is significantly more effective that *Lead interleaved* with *CosineSimilarityHy*, and further, *LexRank* shows a significant degradation in ROUGE-4 effectiveness ("✗"). From this, we conclude that the 5 standard baselines, over DUC 2004, may be weak baselines to use in future experiments, with any claimed improvements questionable.

6 Reproducing Standard and State-of-the-art Algorithms

In this section, we reproduce standard summarisation baselines, and state-of-the-art systems, making several observations over the DUC 2004 and TAC 2008 datasets. In particular, we reimplement the *LexRank*, *Centroid*, *FreqSum*, *TsSum*, and *Greedy–KL* standard baselines. Additionally, we investigate the state-of-the-art summarisation algorithms, that use integer linear programming (ILP) and machine learning techniques, reimplementing *ICSISumm*, and training a supervised machine learned model. Further, we investigate the optimisation of the selection of anti-redundancy components on a per topic basis, making observations regarding the best and worse cases, over DUC 2004 and TAC 2008, for our most effective reimplementations of the standard baselines.

Table 4 provides reference results for standard baselines and state-of-the-art systems, over DUC 2004 and TAC 2008, to which we compare our reimplementations of the various summarisation algorithms. In Table 4, the standard baselines and state-of-the-art reference results, over DUC 2004, are computed from SumRepo. The TAC 2008 reference results are computed from the participants submissions to TAC 2008, specifically *ICSISumm*, which were the most effective runs under ROUGE-2 for part A of the task (the non-update part). Table 5 presents results, over DUC 2004 and TAC 2008, that show the effectiveness of our reimplementations of the 5 standard baselines, our reimplementations of *ICSISumm*, and the machine learned model, *MaxEnt*.

From Table 5, we first observe the ROUGE results for our reimplementations of the standard baselines, where the standard baselines have been numbered (1) to (5). In Table 5, a "✔" symbol indicates a statistically significant improvement of a baseline reimplementation over the standard baseline, while a "†" symbol indicates that there is no statistically significant difference to *ICSISumm* over DUC 2004, and a "‡" symbol indicates no statistically significant difference to *ICSISumm* over TAC 2008. Over DUC 2004, under the target metric ROUGE-2, *GraphPRpriorsHy_CosineSimilarityHy*, *SimCentroidHy_NewWordCount*, and *KLDivergence_CosineSimilarityHy* exhibit improvements over the standard baselines of *LexRank*, *Centroid*, and *Greedy–KL*, respectively, and these 3 baseline reimplementations exhibit similar effectiveness to a state-of-the-art algorithm, *ICSISumm*. We also note further improvements and state-of-the-art effectiveness for our baseline reimplementations under the ROUGE 1 and 4 metrics. For TAC 2008, we observe that reimplementations of *LexRank*, *TsSum*, and *Greedy–KL* exhibit ROUGE-1 effectiveness that is not statistically significantly different from *ICSISumm*.

The improvements for our reimplementations (optimising the standard baselines and closing the gap to the state-of-the-art) are attributed to variations in algorithm design.

Table 4: Reference ROUGE results, for baselines and state-of-the-art systems.

DUC 2004			
Baselines	R-1	**R-2**	R-4
(1) *Cont. LexRank*	36.00	7.51	0.83
(2) *Centroid*	36.42	7.98	1.20
(3) *FreqSum*	35.31	8.12	1.00
(4) *TsSum*	35.93	8.16	1.03
(5) *Greedy–KL*	38.03	**8.56**	1.27

DUC 2004			
State-of-the-art	R-1	**R-2**	R-4
CLASSY 04	37.71	9.02	1.53
CLASSY 11	37.21	9.21	1.48
Submodular	39.23	9.37	1.39
DPP	39.84	9.62	1.57
OCCAMS_V	38.50	9.75	1.33
RegSum	38.60	9.78	1.62
ICSISumm	38.44	**9.81**	1.74

TAC 2008			
State-of-the-art	R-1	**R-2**	R-4
ICSISumm (13)	37.79	11.03	2.26
ICSISumm (43)	38.31	**11.13**	2.20

Table 5: Reimplementation ROUGE results, for baselines and state-of-the-art systems.

DUC 2004			
Baseline Reimplementation	R-1	**R-2**	R-4
(3) *Probability_NewWordCount*	37.52✔†	8.70	1.14
(4) *TopicWordsWp_CosineSimilarityTf*	37.54✔†	8.87	1.39✔†
(1) *GraphPRpriorsHy_CosineSimilarityHy*	38.05✔†	9.34✔†	1.44✔†
(2) *SimCentroidHy_NewWordCount*	37.79✔†	9.37✔†	1.59†
(5) *KLDivergence_CosineSimilarityHy*	38.44†	**9.59**✔†	1.56†
State-of-the-art Reimplementation	R-1	**R-2**	R-4
ILP_ICSISumm	37.77†	9.50†	1.56†
MaxEnt_NewBigrams	38.43†	**9.56**†	1.73†

TAC 2008			
Baseline Reimplementation	R-1	**R-2**	R-4
(3) *Probability_NewBigrams*	35.30	8.05	1.57
(4) *TopicWordsWp_NewWordCount*	36.92‡	9.27	1.93‡
(5) *KLDivergence_CosineSimilarityHyRt*	37.48‡	9.67	2.01‡
(2) *SimCentroidHy_NewWordCount*	36.92	9.77	2.16‡
(1) *GraphDegreeHyRt_NewWordCount*	37.42‡	**10.23**	2.22‡
State-of-the-art Reimplementation	R-1	**R-2**	R-4
MaxEnt_CosineSimilarityRt	36.38	9.51	2.04‡
ILP_ICSISumm	37.31‡	10.24‡	2.20‡

For example, most of the standard baselines use a cosine similarity anti-redundancy component [7], and altering the choice of anti-redundancy component can lead to improvements in effectiveness. Further, the most effective standard baseline reimplementation (over DUC 2004), *KLDivergence_CosineSimilarityHy*, is a variation of *Greedy–KL*. For this reimplementation, instead of greedily selecting the sentences that minimise divergence, our variation first scores sentences by their Kullback–Leibler divergence to all other sentences, then passes the ranked list to an anti-redundancy component. Other variations include altering the vector weighting scheme, such as the hybrid *tf*idf* vectors used by the *SimCentroidHy* baseline reimplementation. From the results presented in Table 5, we conclude that it is possible to optimise the standard baselines, even to the point where they exhibit similar effectiveness to a state-of-the-art system (*ICSISumm*).

Next, from Table 4 and Table 5, we observe that our reimplementation of *ICSISumm*, and the machine learned model *MaxEnt*, exhibit state-of-the-art effectiveness over DUC 2004. In particular, the ROUGE-2 results from our reimplementations of *ICSISumm* and *MaxEnt* are not statistically significantly different from the reference results for the original *ICSIsumm*. Over TAC 2008, we observe similar results with our reimplementation of *ICSISumm*, in that it exhibits effectiveness that is not statistically significantly different to the original. However, we note that the learned model, trained on DUC 2002, is not as effective under ROUGE-2 over TAC 2008 as we observe over DUC 2004. From the results in Table 5, we conclude that our reimplementation of *ICSISumm* is correct, and, although our learned model performs effectively over DUC 2004, the learned model has not generalised effectively from DUC 2002 newswire to TAC 2008 newswire.

We now investigate the manual selection of the most effective anti-redundancy component, on a per topic basis. Taking effective standard baseline reimplementations, we compute ROUGE scores for an oracle system that selects the particular anti-redundancy component, per topic, which maximises the ROUGE-2 effectiveness. Figure 3 visualises the distribution of ROUGE-2 scores, over the 50 topics of DUC 2004, for *KLDivergence_Lead* (no anti-redundancy filtering), and for the oracle system (best case), and additionally, the worst case (where the least effective anti-redundancy component is al-

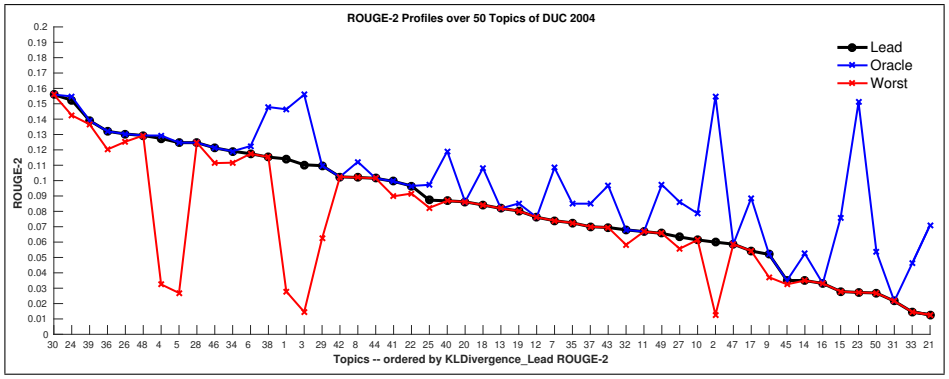

Fig. 3: ROUGE-2 effectiveness profiles, over DUC 2004, for KLDivergence_Lead, an oracle system optimising selection of anti-redundancy components over topics, and the worst case.

Table 6: Results over DUC 2004 and TAC 2008, showing the best/worst scores possible when manually selecting the most/least effective anti-redundancy components per-topic.

DUC 2004			
KLDivergence	R-1	**R-2**	R-4
CosineSimilarityHy	38.44	**9.59**	1.56
Oracle Score	39.06†	9.82	1.70†
Worst Case	35.28	8.04	1.27

TAC 2008			
GraphDegreeHyRt	R-1	**R-2**	R-4
NewWordCount	37.42	**10.23**	2.22
Oracle Score	41.32†	13.17†	4.07†
Worst Case	31.14	6.46	0.82

ways chosen, per topic). Table 6 provides the ROUGE results for *KLDivergence* over DUC 2004, and *GraphDegreeHyRt* over TAC 2008, showing the most effective anti-redundancy component, the effectiveness of the oracle system, and the worst case.

From Figure 3, we can observe that there exists best, and worst case, anti-redundancy component selection choices, per topic. This means, there are topics where we would wish to avoid a particular anti-redundancy component, and further, some topics where we would indeed wish to select a particular anti-redundancy component. If we create an oracle system that manually selects from the 6 different anti-redundancy components, optimising the ROUGE-2 metric over topics, we obtain the ROUGE scores we present in Table 6. From Table 6, we observe that the worst case is always significantly the least effective, over both DUC 2004 and TAC 2008. Further, from Table 6 we observe that the oracle system leads to statistically significant improvements over the most effective anti-redundancy component, indicated by the "† symbol. In particular, over DUC 2004, the oracle system is more effective under ROUGE-1 and ROUGE-4 than the most effective anti-redundancy component (shown in bold). Over TAC 2008, the oracle system is more effective under all ROUGE metrics than the most effective anti-redundancy component (again, shown in bold). From the results in Table 6, we conclude that, while we do not propose a solution for how such an oracle system might be realised in practice, approximations of the oracle system can potentially offer statistically significant improvements in summarisation effectiveness.

7 Conclusions

In this paper, we have reproduced, validated, and generalised findings from the literature. Additionally, we have reimplemented standard and state-of-the-art baselines, making further observations from our experiments. In conclusion, we have confirmed that the ROUGE-2 metric is aligned with crowd-sourced user judgements for summary quality, and confirmed that several state-of-the-art systems behave differently, despite similar ROUGE-2 scores. Further, an optimal *Lead* baseline can be derived from interleaving

the first sentences from multiple documents, and applying anti-redundancy components. Indeed, an optimal *Lead* baseline exhibits ROUGE-2 effectiveness with no significant difference to standard baselines, over DUC 2004. Additionally, the effectiveness of the standard baselines, as reported in the literature, can be improved to the point where there is no significant difference to the state-of-the-art (as illustrated using *ICSISumm*). Finally, given that an optimal choice of anti-redundancy components, per-topic, exhibits significant improvements in summarisation effectiveness, we conclude that future work should investigate learning algorithm (or topic) specific anti-redundancy components.

Acknowledgements

Mackie acknowledges the support of EPSRC Doctoral Training grant 1509226.
McCreadie, Macdonald and Ounis acknowledge the support of EC SUPER project (FP7-606853).

References

[1] Allan, J., Wade, C., Bolivar, A.: Retrieval and Novelty Detection at the Sentence Level. In: Proc. of SIGIR 2003.
[2] Conroy, J.M., Schlesinger, J.D., O'Leary, D.P.: Topic-focused Multi-document Summarization Using an Approximate Oracle Score. In: Proc. of COLING-ACL 2006.
[3] Erkan, G., Radev, D.R.: LexRank: Graph-based Lexical Centrality as Salience in Text Summarization. Journal of Artificial Intelligence Research 22(1) (2004).
[4] Gillick, D., Favre, B.: A Scalable Global Model for Summarization. In: Proc. of ACL ILP-NLP 2009.
[5] Guo, Q., Diaz, F., Yom-Tov, E.: Updating Users about Time Critical Events. Advances in Information Retrieval 7814 (2013).
[6] Haghighi, A., Vanderwende, L.: Exploring Content Models for Multi-document Summarization. In: Proc. of NAACL-HLT 2009.
[7] Hong, K., Conroy, J., Favre, B., Kulesza, A., Lin, H., Nenkova, A.: A Repository of State of the Art and Competitive Baseline Summaries for Generic News Summarization. In: Proc. of LREC 2014.
[8] Kedzie, C., McKeown, K., Diaz, F.: Predicting Salient Updates for Disaster Summarization. In: Proc. of ACL-IJCNLP 2015.
[9] Lin, C.Y.: ROUGE: a Package for Automatic Evaluation of Summaries. In: Proc. of ACL 2004.
[10] Lin, C.Y., Hovy, E.: The Automated Acquisition of Topic Signatures for Text Summarization. In: Proc. of COLING 2000.
[11] Mackie, S., McCreadie, R., Macdonald, C., Ounis, I.: Comparing Algorithms for Microblog Summarisation. In: Proc. of CLEF 2014.
[12] Mackie, S., McCreadie, R., Macdonald, C., Ounis, I.: On Choosing an Effective Automatic Evaluation Metric for Microblog Summarisation. In: Proc. of IIiX 2014.
[13] McCreadie, R., Macdonald, C., Ounis, I.: Incremental Update Summarization: Adaptive Sentence Selection based on Prevalence and Novelty. In: Proc. of CIKM 2014.
[14] Nenkova, A.: Automatic Text Summarization of Newswire: Lessons Learned from the Document Understanding Conference. In: Proc. of AAAI 2005.
[15] Nenkova, A., McKeown, K.: Automatic Summarization. Foundations and Trends in Information Retrieval 5(2-3) (2011).
[16] Nenkova, A., Vanderwende, L., McKeown, K.: A Compositional Context Sensitive Multi-document Summarizer: Exploring the Factors That Influence Summarization. In: Proc. of SIGIR 2006.
[17] Owczarzak, K., Conroy, J.M., Dang, H.T., Nenkova, A.: An Assessment of the Accuracy of Automatic Evaluation in Summarization. In: Proc. of NAACL-HLT WEAS 2012.
[18] Radev, D.R., Jing, H., Styś, M., Tam, D.: Centroid-based Summarization of Multiple Documents. Information Processing & Management 40(6) (2004).
[19] Spärck Jones, K.: Automatic Summarising: The State-of-the-art. Information Processing & Management 43(6) (2007).